Information Sciences 198 (2012) 48–61

Feature selection using structural similarity

Sushmita Mitra [a], Partha Pratim Kundu [a,*], Witold Pedrycz [b,c]

[a] Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700 108, India
[b] Electrical & Computer Engg., University of Alberta, Edmonton, Canada T6G 2G7
[c] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

Keywords:
Structural similarity
Multi-objective optimization
Feature selection
Proximity
Membership

ABSTRACT

A new method of feature selection is developed, based on structural similarity. The topological neighborhood information about pairs of objects (or patterns), to partition(s), is taken into consideration while computing a measure of structural similarity. This is termed proximity, and is defined in terms of membership values. Multi-objective evolutionary optimization is employed to arrive at a consensus solution in terms of the contradictory criteria pair involving fuzzy proximity and feature set cardinality. Results for real and synthetic datasets, of low, medium and high dimensionality, show that the method led to a correct selection of the reduced feature subset. Comparative study is also provided, and quantified in terms of accuracy of classification and clustering validity indices.

1. Introduction

Feature selection is essential in analyzing large data, particularly being a preprocessing step for reducing dimensionality, removing irrelevant features, reducing storage requirements and enhancing output comprehensibility. It is a process that selects a minimum subset of n' features from an original set of n features ($n' < n$), such that the feature space is optimally reduced according to a certain predetermined evaluation criterion. This often involves selecting subsets of features useful to build good predictors [16].

Search is a key issue in feature selection, involving search starting point, search direction, and search strategy. One also needs to measure the goodness of the generated feature subset. Feature selection can be supervised as well as unsupervised, depending on class information availability in data. The algorithms are typically categorized under filter and wrapper models [23], based on whether or not the learning methodology is used to select the feature subset. The wrapper methods assess feature subsets according to their usefulness to a given predictor. However selecting a good set of features is usually suboptimal for building a predictor, particularly in the presence of redundant variables. Since finding the best feature subset is found to be intractable or NP-hard [1], therefore heuristic and non-deterministic strategies are deemed to be practical.

Feature selection can be supervised or unsupervised. Supervised feature selection mostly depends on the performance of a chosen classifier. In the absence of class information, the unsupervised techniques use some intrinsic property of the data [26]. Here, no external information like class label of an instance is needed. Related literature on feature subset evaluation include Category Utility score [9], Fisher's feature dependency measure [37,11], entropy based unsupervised feature ranking [6], and generally proceed by selecting the subset(s) of features while preserving the inherent characteristic of data. In Ref. [38], the authors use an unsupervised method that assumes a linear model to choose a subset of features which can approximate the original data. Zhao et al. [39] have proposed an embedded model which evaluates a feature subset based on their capability of preserving sample similarity.

The use of soft computing is an interesting proposition along this direction [2], in order to arrive at an acceptable solution at a lower cost. This is of particular interest towards the efficient mining and analysis of large data. We can utilize the uncertainty handling capacity of fuzzy sets [25] and the search potential of genetic algorithms for efficiently traversing large search spaces. When there are two or more conflicting characteristics to be optimized, often the single objective optimization function requires an appropriate formulation in terms of an additive combination of the different criteria involved. In such cases a *multi-objective* optimization becomes more appropriate. Multi-objective GAs (MOGAs) [7] may be used as a tool, while efficiently searching for optimal solutions.

An interesting way of looking at feature selection is to aim at preserving the structural similarity of data clusters, while mapping a high-dimensional feature space to a lower-dimensional one. In other words, a pair of objects (or patterns) belonging to the same partition in the original high-dimensional space is expected to be retained in the same partition in the reduced domain as well. By considering such similarity or proximity between all object pairs as a guideline [29], one can hope to eliminate some of the less important features. The aim is to retain those features which allow the similarity between the partitioning, in the original and reduced spaces, to be high. This can also help in improving the computational efficiency in the lower dimensional space, given that the mapping is nearly lossless as measured in terms of the similarity measure used.

In this article, we propose such a method of feature selection, based on structural similarity. The topological neighborhood information about pairs of objects (or patterns), to partition(s), is taken into consideration while computing a measure of structural similarity. This is termed proximity, and is defined in terms of membership values of the corresponding patterns. For a dataset with N input patterns we can define an $N \times N$ symmetric matrix, referred as proximity matrix P, whose (i,j) th entry represents the similarity (or dissimilarity) measure for the ith and jth patterns for $i, j = 1, \ldots, N$. Typically distance functions are used for the purpose. The proximity matrix is a pertinent construct that allows us to deal with structural information inherent in the data. In the fuzzy perspective the concept of similarity boils down to the membership values.

We focus on the use of proximity relationship, as a similarity measure, from the viewpoint of fuzzy sets. This is used as one of the objective functions, during multi-objective optimization, for evaluating the fitness of the feature subsets of varying cardinality. The use of fuzziness allows us to efficiently model uncertainties and ambiguities inherent in real life overlapping data. The proximity of a pair of patterns in the original feature space is compared with that in the reduced subspace of selected features. If they are similar, as measured in terms of their belonging to the same cluster (both before and after feature selection), then this implies that the eliminated feature(s) are not so relevant to the decision making process. The second criterion is the cardinality of the selected feature subset. This is sought to be minimized, and serves as a penalty to the objective function. A close observation reveals that these two criteria are of a conflicting nature. A smaller subset of features is likely to result in a reduced proximity, and hence reduced classification accuracy (as compared to the original feature space).

Multi-objective optimization is employed to arrive at a consensus solution in terms of this contradictory criteria pair, involving fuzzy proximity and feature set cardinality. Here MOGA is used as a tool for the multi-objective optimization, and any other technique could also have sufficed. The user does not need to specify the desired number of features, as it is embedded in the optimization process. The algorithm terminates when an optimal subset of features is obtained, according to the fitness criteria of the multi-objective genetic optimization. Experimental results indicate correct selection of the reduced feature subset. Validation of the selected set of features is reported in terms of classification accuracy using WEKA [17] implementation of several well-known classifiers, as well as internal and external clustering validity indices.

The rest of the paper is organized as follows. In Section 2 we present the proximity-based methodology for feature selection and outline the background on multi-objective optimization. The experimental results and comparative study are described in Section 3, on various real and synthetic datasets. Finally, Section 4 concludes the article.

2. Proximity-based feature selection

Let us consider Fig. 1 to explain the concept of structural similarity between clusters in the context of feature selection. Using this crude example, we have discussed that the idea of preserving cluster structure of original feature space in a feature subset, would actually lead to feature selection. Removing irrelevant feature(s) does not affect much the internal characteristics of data. Three patterns $X1$, $X2$ and $X3$ are seen to be partitioned into the same cluster in the three-dimensional feature space of part (a). The three features are aligned with three reference axes *i.e.* x-axis, y-axis and z-axis of this dataset. If the least important feature *i.e.* the feature aligned with y-axis is eliminated, the cluster structure is expected to remain unaltered; implying that the single cluster would still contain the same distribution of pattern points as depicted in part (b) of the figure. Here the three- to two-dimensional mapping is said to be almost lossless, such that the clustering structures in the two subspaces are very similar. The clustering structure is said to be preserved in the transformation between the two subspaces. On the other hand, if an important feature *e.g.* the feature aligned with z-axis is eliminated then the mapping is bound to disrupt the cluster structure since important information gets lost in the process. From part (c) of the figure we observe that the similarity between the partitioning, in the two subspaces, is now no longer high. In other words, the distance between the partitioning is higher; with the pattern points getting redistributed into two different clusters *i.e.* cluster structure of original space is not preserved here.

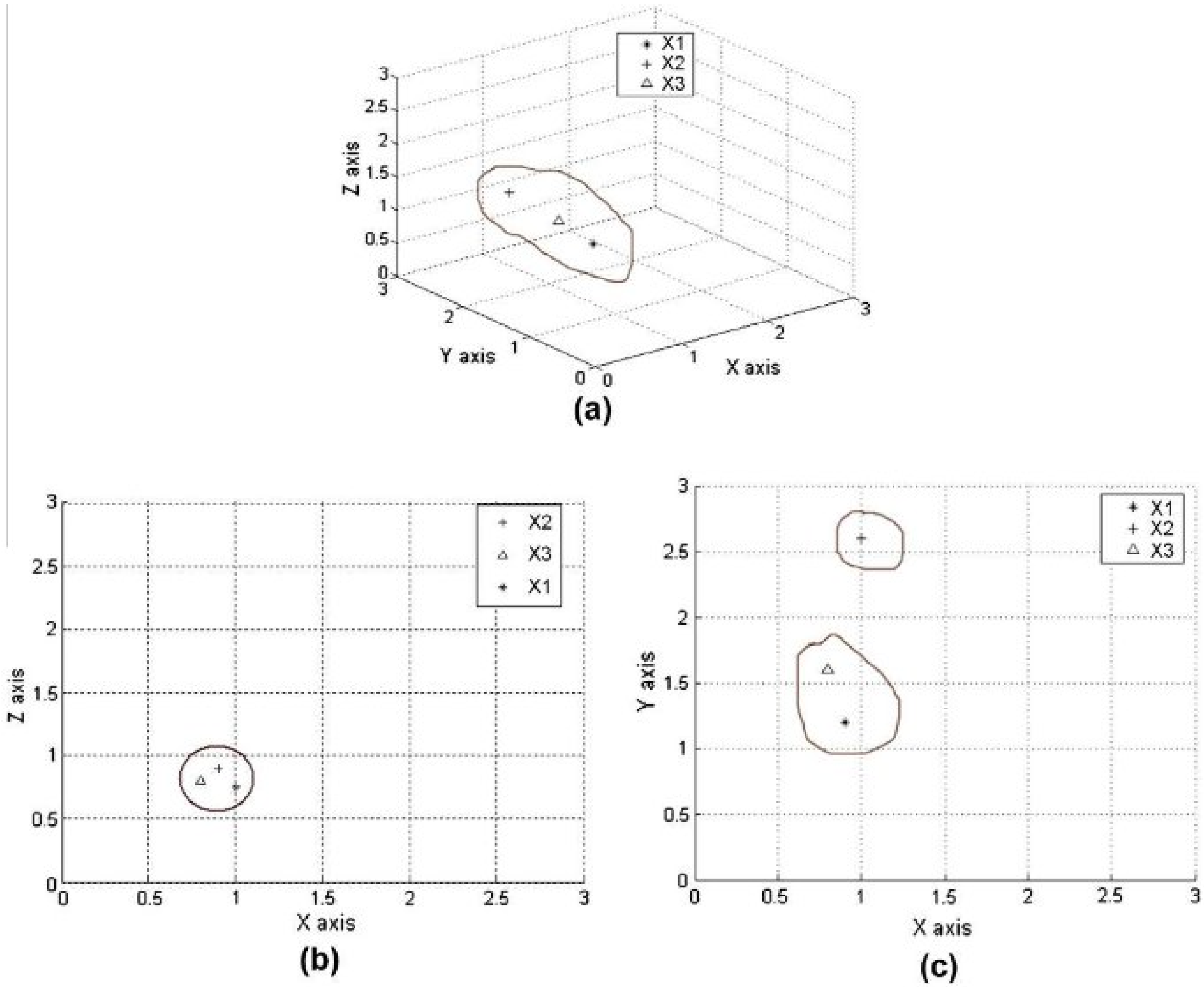

Fig. 1. Mapping of patterns from (a) three-dimensional space, to two two-dimensional spaces having cluster structure (b) preserved, and (c) not preserved.

In this paper we use proximity as a way of determining the similarity between clustering structures, while mapping from a high- to a low-dimensional feature space. In the process, we aim to retain the important features. Such preservation of structural similarity between clusters is expected to lead to the selection of important features. Let there be K subsets of data located in different feature subspaces, with the number of patterns in each subspace being equal to N. We form a $c \times N$ partition matrix $\mathcal{P}$ consisting of membership values μ_{ij}. This membership value is updated by minimizing an objective function J_m defined in Eq. (1) [4].

$$J_m = \sum_{j=1}^{N}\sum_{i=1}^{c}\mu_{ij}^{m'}\|x_{ij} - m_j\|^2 \tag{1}$$

We compute $\mu_{ij} \in [0,1]$ as the membership of the jth pattern to the ith mean m_i, where $\|.\|$ is the distance norm and $1 \leqslant m' < \infty$ is the fuzzifier [4]. Note that the dimensionality n of the patterns in each subset could be different. However, in each subset, the distance of a pattern is computed from the fuzzy cluster prototypes over the same set of features. We have

$$m_i = \frac{\sum_{j=1}^{N}(\mu_{ij})^{m'}x_j}{\sum_{j=1}^{N}(\mu_{ij})^{m'}} \tag{2}$$

and

$$\mu_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{d_{ji}}{d_{jk}}\right)^{\frac{2}{m'-1}}}, \tag{3}$$

$\forall i$, with $d_{ji} = \|x_j - m_i\|^2$, subject to $\sum_{i=1}^{c}\mu_{ik} = 1$, $\forall k$, and $0 < \sum_{k=1}^{N}\mu_{ik} < N$, $\forall i$. Typically, we choose $m' > 1$.

The partition matrix is used to evaluate proximity p, which measures the extent to which a pair of patterns are regarded as similar or dissimilar in different subspaces [29]. This incorporates a mechanism of partial supervision in the process of navigating a structure in the data. The proximity matrix P contains the proximity results for all possible pairs of patterns. The fuzzy partitions conveyed by Eq. (3), obtained by running fuzzy c-means (FCM) [4], are directly related to the proximity relation. The proximity between pattern pair k_1 and k_2 is computed as

$$p(k_1,k_2) = \sum_{i=1}^{c}(\mu_{ik_1} \wedge \mu_{ik_2}), \tag{4}$$

where $\wedge$ denotes the minimum operation, $p(k_1,k_2) \in [0,1]$, and k_1, $k_2 = 1, \ldots, N$. Evidently $p(k_1,k_2) = 1$ for $k_1 = k_2$ such that membership is evaluated with respect to FCM, and $p(k_1,k_2) = p(k_2,k_1)$.

The aim is to reduce the number of features, subject to maintaining the structural similarity between patterns. For this purpose we employ multi-objective optimization to handle the conflicting requirements of dimensionality reduction along with proximity preservation. We use genetic algorithm (MOGA) [here NSGA-II] [8], as a tool to efficiently traverse the feature subspaces, subject to fulfilling the above objectives.

2.1. Proximity between feature subspaces

Let the cardinality of the original and reduced feature spaces be n and n', respectively. Let the proximity matrices in these two spaces be denoted by P and P'. The similarity between the two matrices is represented by a scalar value

$$P_s = \sum_{k_1=1,k_2>k_1}^{N}[p(k_1,k_2) \wedge p'(k_1,k_2)], \tag{5}$$

where $p'(k_1,k_2)$ is computed by Eq. (4) in the reduced feature space and $\wedge$ denotes the minimum operation.

Note that the membership value μ_{ik} at each stage is computed based on the FCM objective function, using Eqs. (2) and (3). This becomes inherent in the proximity matrix in Eq. (4). Moreover, as the MOGA updates the encoded cluster means over the generations it has to continuously refer to the FCM based membership computations.

We retain only those pattern pairs which belong to the same cluster in both the original and the reduced feature space, in an attempt to reduce the ambiguity of the resultant clustering. For such cases we use

$$P_{s_0} = \sum_{k_1=1,k_2=k_1+1}^{N}[(p(k_1,k_2) \geqslant \theta) \wedge (p'(k_1,k_2) \geqslant \theta)], \tag{6}$$

such that P_{s_0} takes the minimum of the values of $p(k_1,k_2)$ and $p'(k_1,k_2)$ only when both $p(k_1,k_2)$ and $p'(k_1,k_2)$ are greater than a threshold θ. This implies that both μ_{ik_1} and μ_{ik_2} are greater than or equal to θ in the original and reduced feature spaces by Eq. (4).

2.2. Evaluation of the subspaces

The resultant partitioning in the different feature subspaces is evaluated both internally and externally. While the external measures compare the resultant partitioning with the correct classification of the (known) data, the internal measures compute a relationship involving the inter- and intra-cluster separability. There exist many measures of this type in literature [20,15,18,14,35]. Some of them are discussed and presented here.

The Silhouette statistic [34] offers one way of internally validating the generated clusters. Though computationally more intensive, it is another way of estimating the number of clusters in a distribution. The Silhouette index, S, computes for each point a width depending on its membership in any cluster. This silhouette width is then an average over all observations. This is expressed as

$$S_k = \frac{1}{N_k}\sum_{i:x_i\in C_k}\frac{b_i - a_i}{\max(a_i,b_i)}, \tag{7}$$

where N_k is the total number of points of cluster C_k, a_i is the average distance between pattern $\mathbf{x}_i$ and all other points in its own cluster C_k, and b_i is the minimum of the average dissimilarities between $\mathbf{x}_i$ and patterns in other clusters. Finally, the global silhouette index, S, of the clustering is given by

$$S = \frac{1}{c}\sum_{k=1}^{c}S_k. \tag{8}$$

The partition with the highest value of S is considered to be optimal.

The F-measure is an external validation technique, using class labels as external information. It combines precision and recall [32], expressed as

$$Recall(i,j) = \frac{n_{ij}}{n_i}, \quad Precision(i,j) = \frac{n_{ij}}{n_j}, \tag{9}$$

where n_{ij} is the number of patterns belonging to class i that fall in cluster j, and n_i, n_j are the cardinalities of class i cluster j respectively. The $F(i,j)$ of cluster j and class i is computed as

$$F(i,j) = \frac{2 \times Recall(i,j) \times Precision(i,j)}{Recall(i,j) + Precision(i,j)}. \tag{10}$$

No one-to-one mapping exists between a class and a cluster. The $F(i)$ for a particular class i is given as

$$F(i) = \max_{0<j<c} F(i,j). \tag{11}$$

Finally, the F-measure is evaluated as

$$F = \sum_i \frac{n_i}{N} F(i), \tag{12}$$

with values lying in the range $[0,1]$, and a larger value of F indicating improved quality of clustering.

To compare partitions in reduced and original feature spaces, some well known external measures like Jaccard coefficient (J) [3], Rand index [31] and one information theoretic measure, Variation of Information (VI) [24] is used. Jaccard and Rand index are ranged in $[0,1]$ and VI in $[0, \log N]$ where N is the total number of points. A value nearer to 1 indicates better matching of the partitions two different spaces in case of J and Rand index and in case of VI, similar partitions generate a value nearer to 0.

The performance of the selected feature subsets, obtained on the basis of maximization of proximity and minimization of cardinality, is also externally validated in terms of its predictive accuracy. This is measured by several well-known classifiers [12], like k-nearest neighbor (k-NN), Naive Bayes [10] and support vector machine (SVM).

2.3. Multi-objective optimization

Multi-objective optimization [7] trades off between a vector of objective functions $\vec{F}(\vec{x}) = F_1(\vec{x}), F_2(\vec{x}), \ldots, F_M(\vec{x})$, where M is number of objectives and $\vec{x}(\in \mathcal{R}^n)$ is a vector of n decision variables. Unlike single-objective optimization problems, here we try to optimize two or more conflicting characteristics represented by objective functions. Modeling this situation in a single objective case would amount to a heuristic determination of a number of parameters involved in expressing such a scalar-combination-type objective function. The multi-objective technique, on the other hand, is concerned with minimization or maximization of a vector of objectives $\vec{F}(\vec{x})$ that can be the subject of a number of constraints or bounds. In other words, we have

$$\begin{array}{rl} \text{Minimize/Maximize} & \vec{F}(\vec{x}) \\ \text{subject to} & g_i(\vec{x}) \leqslant 0, \quad i = 1, 2, \ldots, I; \\ & h_k(\vec{x}) = 0, \quad k = 1, 2, \ldots, K; \\ & x_j^L \leqslant x_j \leqslant x_j^U, \quad j = 1, 2, \ldots, n; \end{array} \tag{13}$$

where I and K are the inequality and equality constraints respectively. Each decision variable x_j takes a value within lower bound x_j^L and upper bound x_j^U, with the bounds constituting a decision variable space $\mathcal{D}$. The solution set of $\vec{x}$ that satisfies *all* $(I + K)$ constraints and *all* $2n$ variable bounds, forms the feasible solution space Ω. As these objective functions are competing with each other, there is no unique solution to this technique. Instead, the concept of non-dominance [7] (also called Pareto optimality [5]) must be used to characterize the objectives. The objective function space Λ is defined as $\Lambda = f \in \mathcal{R}^m$, where $f = \vec{F}(\vec{x})_{\vec{x} \in \Omega}$. A mapping from feasible solutions space into objective function space, in two dimensions, is depicted in Fig. 2

The concept of optimality, behind the multi-objective optimization, deals with a set of solutions. The conditions for a solution to be *dominated* with respect to the other solutions are outlined here. A solution $\vec{x}^{(1)}$is said to dominate the other solution $\vec{x}^{(2)}$ if the following two conditions are true [7]:

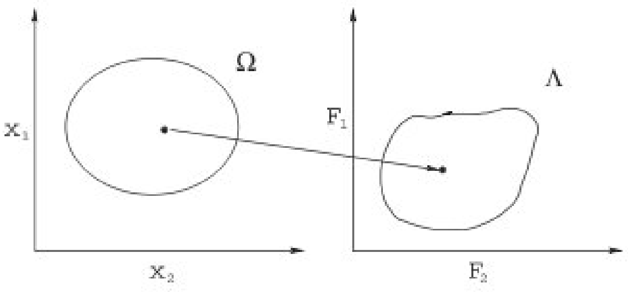

Fig. 2. Mapping from feasible solutions space into objective function space.

1. The solution $\vec{x}^{(1)}$ is *no worse* than $\vec{x}^{(2)}$ in all M objectives, *i.e.*

   $F_i(\vec{x}^{(1)}) \ntriangleright F_i(\vec{x}^{(2)}) \quad \forall i = 1, 2, \ldots M.$

2. The solution $\vec{x}^{(1)}$ is *strictly better* than $\vec{x}^{(2)}$ in *at least one* of the M objectives, *i.e.*

   $F_{\bar{i}}(\vec{x}^{(1)}) \triangleleft F_{\bar{i}}(\vec{x}^{(2)})$ for at least one $\bar{i} \in \{1, 2, \ldots M\}$

If any of the above conditions is not satisfied, then the solution $\vec{x}^{(1)}$ does not dominate the solution $\vec{x}^{(2)}$. So, the solution $\vec{x}^{(1)}$ and $\vec{x}^{(2)}$ form Pareto optimal front of these objective functions. A typical Pareto-optimal front over two objective functions is shown in Fig. 3. Here we simultaneously optimize the conflicting requirements of the multiple objective functions. Genetic algorithms may be used as a tool for multi-objective optimization. In this article we have used the Non-dominated Sorting Genetic Algorithm (NSGA-II), that has been shown to convergence to the global Pareto front while simultaneously maintaining the diversity of the population [8].

2.4. Optimization tool

The multi-objective optimization is implemented using MOGA, specifically NSGA-II. We encode the problem as a real-coded string of length L, with the first n bits corresponding to the n features in the original space. Here, in the bit representation, a "1" implies that the corresponding attribute is present while "0" indicates that it is not. The desired number of features need not be pre-specified, since it is automatically determined during the optimization. Let the size of a chromosome be

$$L = n + c \times n = n \times (c + 1). \tag{14}$$

The c cluster centers (or prototypes) are encoded in real form in the subsequent $c \times n$ bits. Only those features of the centers in the second part of the string, corresponding to a "1" in the first part, are considered during clustering. Fig. 4 depicts such an encoding in a chromosome, representing a sample set of cluster prototypes in a feature subspace. Initially all the bits are set randomly.

The objective is to optimize a conflicting set of requirements; *i.e.*, select a minimal number of features that enable us to arrive at an acceptable structure-preserving mapping. We employ MOGA with P_{s_0} of Eq. (6) as the fitness function

$$f_1 = P_{s_0}. \tag{15}$$

The second fitness function corresponds to the cardinality of the feature set under consideration, and is defined as

$$f_2 = n'. \tag{16}$$

While f_2 is minimized to give credit to a candidate string containing less attributes, the function f_1 maximizes the extent to which all pairs of patterns belong to the same cluster in the two feature spaces, *viz.*, original and reduced subspace. These two fitness functions are optimized in the framework of MOGA. Clustering is done by FCM to update the prototypes m_i, in the different subspaces.

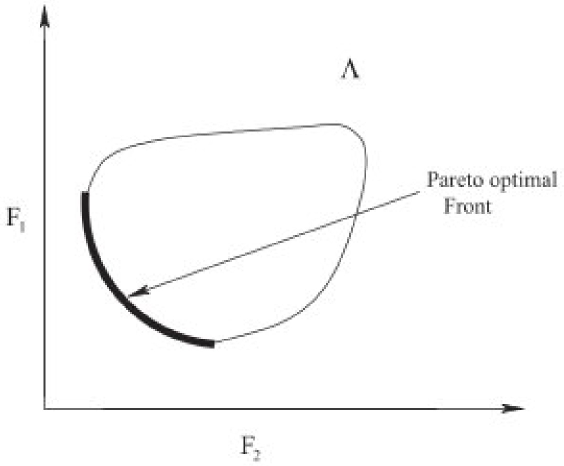

Fig. 3. Pareto optimal front or non-dominated solutions of F_1 and F_2.

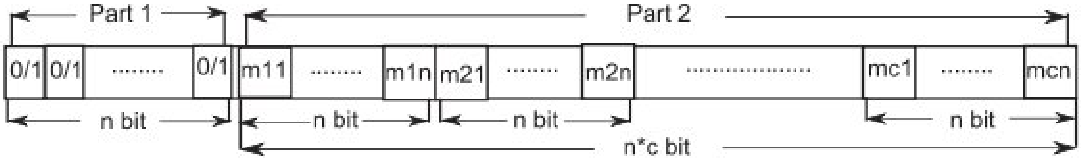

Fig. 4. An encoded chromosome representing a feature subspace with the cluster prototypes.

2.5. The algorithm

The objective is to preserve the proximity relationship between pattern pairs, which is a measure of their structural similarity, while reducing the number of features. The main steps of the algorithm, outlined below, are repeated over several generations.

1. Initialize the population randomly, with real numbers.
2. Select a pair of chromosomes randomly for single-point crossover.
3. Perform two-point mutation simultaneously on the two parts of the string. In the first part, the value of the randomly chosen bit (signifying presence or absence of the corresponding attribute) is flipped. In case of the second part, the value $m_{ij_{old}}$ corresponds to the randomly chosen attribute j of the ith cluster center; this is mutated as

$$m_{ij_{new}} = \sigma \times x + m_{ij_{old}}, \tag{17}$$

where the perturbation $x \sim Norm(0,1)$ is drawn from a Gaussian distribution, the variance σ^2 determines the magnitude of this perturbation at position $m_{ij_{old}}$, and $m_{ij_{new}}$ is its new value (at the corresponding attribute j of the ith cluster center) after mutation.
4. Calculate the fitness values of different feature sets based on their proximity and cardinality, using Eqs. (15) and (16).
5. Rank the population using dominance criteria. Calculate the crowding distance of the chromosome, to maintain diversity in the population [8].
6. Combine parent and offspring population. Replace the parent population by the best members of the combined population.

Note that the cluster centers are initially set randomly. During crossover and mutation the centers get modified. Their effect is reflected through the proximity function (Eq. (15)) into the fitness evaluation. The features present in a chromosome, as indicated by the "1"s in the first part, determine the reduced feature subspace. They affect the computation of proximity in terms of cluster prototypes, using Eqs. (2)–(4). Finally the selected feature sets are validated in terms of cluster validity indices (Eqs. (8) and (12)) and the classification accuracy.

3. Experimental results

The performance of the algorithm was tested on various synthetic and real datasets. These include (i) a synthetic dataset and the benchmark *Iris* flower (low-dimensional), (ii) *Ionosphere* and *Spambase* (medium-dimensional), and (iii) *Isolet* and *Colon* cancer microarray gene expression data (high-dimensional). All results were averaged over several (3–5) runs involving different random seeds. No significant change was observed in the performance, using different seeds. The choice of θ in Eq. (6) was taken to be 0.5, so that the membership of pattern pair k_1, k_2 became simultaneously high in the same cluster. The crossover and mutation probabilities, in the MOGA, were selected as 0.85 and 0.05 respectively after several experiments. The clustering was evaluated in terms of clustering validity indices. The selected feature subsets were externally validated, using the publicly available WEKA implementation [17] of different classifiers like k-nearest neighbors (k-NN), Naive Bayes' (NB) and support vector machine (SVM), involving ten-fold cross-validation. The clustering structures of reduced and original features space are compared using J, Rand index and VI.

3.1. Data description

The synthetic data contains three clusters, each with 100 randomly generated patterns. The two-dimensional scatter plot of Fig. 5 depicts the patterns lying within circles of unit radius, each having different centers. A lot of overlapping is artificially introduced. We introduced a third attribute having completely random values, to evaluate the effectiveness of the algorithm in identifying the significance of the first two features. The *Iris* data [13] consists of 150 pattern points with four input features corresponding to measurements of *sepal length, sepal width, petal length, petal width* on fifty flowers from each of three species *setosa, versicolor, virginica* (representing the three output classes).

The *Ionosphere* data represents autocorrelation functions of radar measurements. There are 351 instances, each having 34 (continuous) features and belonging to two classes, *viz.* "good" or "bad" – indicating the passage or obstruction of free electrons in the ionosphere. We considered a total of 32 features (attributes 3–34) as input to the algorithm. The *Spambase* data

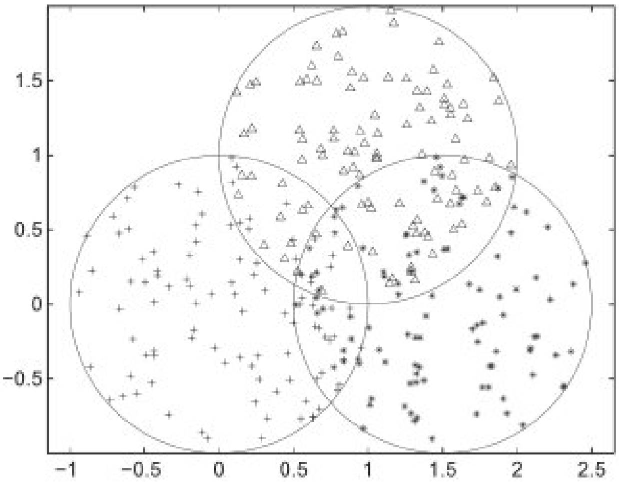

Fig. 5. Synthetic data.

consists of 4601 instances of emails, to be classified into spam or non-spam categories. There are 57 continuous attributes denoting word frequencies.

The *Isolet* data consists of several spectral coefficients of utterance of English alphabets by 150 subjects. There are 617 real features (having values in the range [0,1]) with 7797 instances and 26 classes. The above-mentioned three datasets were taken from the UCI Machine Learning Repository.[1]

The *Colon Cancer* data[2] is a collection of 62 gene expression measurements from colon biopsy samples. There are 22 normal and 40 colon cancer samples, having 2000 genes (features). Typically, microarray gene expression data involves a larger number of features (genes) as compared to the samples (time points). In other words, the features correspond to gene expression values that indicate the abundance of mRNA in a sample (or tissue) for a number of patients; with the objective being to separate cancer patients from healthy ones based on their gene expression profiles. Many of these features are redundant and adversely affect the output decision. Hence preprocessing is often needed [16] to initially eliminate some of the irrelevant features. Some initial preprocessing [2] was done, to reduce the large number of redundant genes to 943, before applying the proposed algorithm.

3.2. Low- and medium-dimensional data

The performance of the algorithm for strings generated in the non-dominated Pareto front, for the four datasets (having low and medium number of features), are presented in Tables 1 and 2. The second column (in both tables) indicates the selected attributes, marked by a "1" in the first part of the chromosome, with the string corresponding to feature positions 1, 2, …, *n*. The two fitness functions are evaluated by Eqs. (15) and (16). However in cases where the original feature space did not figure in the Pareto optimal front, this is still included as the last row for each dataset in the table for comparison (without any f_1). The external validation performance of the selected feature subsets is provided, along with that of the original set, in terms of classification accuracy involving ten-fold cross-validation using different classifiers. The algorithm was run for 100 generations with a population size of 50 chromosomes. The Silhouette index (Eq. (8)) and F- measure (Eq. (12)) values are listed under the column heading Silh. Stat. (S) and F-meas. respectively.

We know that the synthetic data is represented with the first two attributes, and the third feature was inserted randomly. As evident from the results, the selection of the first two features (only) generally results in the best overall accuracy, as well as S and F, due to the elimination of this unimportant third feature. The feature set {1,2} also produces better clustering in reduced space according to J, Rand index and VI.

In case of the *Iris* data, it is observed that the choice of feature 3 occurs in all the three cases, with feature 4 being selected the second-most frequently. Together they result in the second highest proximity and second lowest cardinality. It is well-known that these are the two features most important for discriminating between the classes in this benchmark data. Interestingly, the performance of the *k*-NN in the reduced space (involving attributes 3 and 4) is found to be the overall best – inspite of the elimination of two features. The same holds for the validity indices S and F. The SVM provides best accuracy with three features while NB performs best with only feature 3. The J, Rand index and VI are shows that the cluster structure is best preserved also in feature 3.

The results from Table 2 exhibit better average classification performance by *k*-NN and SVM, for *Spambase*, with a smaller set of features *viz.* 13 and 15. The values of both S and F are also the best with 15 features. Although NB provides a better

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html.

[2] http://microarray.princeton.edu/oncology.

Table 1
Performance of selected feature subsets, of low cardinality, from Pareto-optimal front.

Dataset	Feature subspace	f_1 (prox. ($\times 10^4$))	f_2 (card.)	Validation accuracy (%) by				Silh. Stat. (S)	F-meas.	J	Rand	VI
				k-NN		NB	SVM					
				k =								
Synthetic				1	77.4							
N = 300	{1, 2, 3}	2.10	3	3	73.9							
n = 3	(Original)			5	76.0	78.0	78.7	0.055	0.395	–	–	–
c = 3				7	77.6							
				1	60.3							
	{1}	0.74	1	3	59.5							
				5	60.2	62	61.3	0.044	0.622	0.21	0.55	2.13
				7	60.3							
				1	**79.2**							
	{1, 2}	1.53	2	3	**79.1**							
				5	**80.4**	**80**	**80.3**	**0.088**	**0.801**	0.20	0.56	2.17
				7	**80.8**							
Iris				1	94.7							
N = 150	{2, 3, 4}	0.58	3	3	94.2							
n = 4				5	93.8	96.0	**97.3**	0.176	0.940	0.80	0.93	0.38
c = 3				7	93.0							
				1	93.0							
	{3}	0.34	1	3	92.9							
				5	92.1	**96.7**	95.3	0.214	0.933	**0.82**	**0.93**	**0.37**
				7	92.4							
				1	**94.8**							
	{3, 4}	0.36	2	3	**94.6**							
				5	**94.2**	96.0	96.7	**0.219**	**0.950**	0.79	0.92	0.42
				7	**94.3**							
				1	93.3							
	(Original)	–	4	3	92.6							
				5	91.4	96.0	96.7	0.156	0.677	–	–	–
				7	89.7							

The significance of bold values represent the best value in a set of measured values.

score of 79.3% in the original space, yet its performance with 15 features is comparable at 79.0%. The *J*, Rand index and *VI* provide best result with 11 features *i.e* the original cluster structure is found on that feature space.

Results for the *Ionosphere* data demonstrate that out of the 32 initial attributes our algorithm selected a cardinality of 5 and 7 for the best performance in terms of mean recognition accuracy (%) by *k*-NN. In Fig. 6 we depict a visually understandable, three-dimensional projection, in terms of attributes 4, 5, 6 of the 32-dimensional data. Incidentally, this corresponds to the best performance by classifier NB. It is observed here that our algorithm selected a reasonably good set of features, which captured the structural similarity between the two classes in the original feature space (at the best values of *S* and *F*). The best feature subset in terms of structure preservation is a set of 16 features according to the *J*, Rand index and *VI*.

Next the scope of the algorithm was extended to incorporate a variation in the number of clusters. We determined the optimum number of clusters c_o (varying c from 2 to 12), in both the original and reduced feature spaces, by maximizing the Silhouette index of Eq. (8). FCM is used to determine the fuzzy partitioning corresponding to the c_o value for each generated feature subspace. Multiobjective optimization in terms of maximization of proximity (Eq. (15)) and minimization of cardinality of the feature space (Eq. (16)) ensures the selection of those feature subsets that retain structural similarity among the clusters. The encoded chromosome of Fig. 4 now involves only the first n bits. However, the computational complexity gets enhanced and adversely affects the processing of large data.

Table 3 depicts the results for the *Iris* and *Ionosphere* data. In all the cases the optimum number of clusters converged to c_o = 2. Incidentally the corresponding value of *S* was found to be better here, as compared to Tables 1 and 2. The algorithm, in this modified framework, generated the same subsets of reduced features in Table 3 as in Table 1. In case of *Synthetic* data the extended algorithm failed to eliminate the random third feature. The *Spambase* data was found to be too large to be processed, upon varying the number of clusters. With the *Ionosphere* data we obtained a different set of reduced feature subsets, that were generally comparable in terms of predictive accuracy and *F*-measure. The result of the *J*, Rand index and *VI* shown that cluster structure is also preserved when we allow variation clusters.

3.3. High-dimensional data

Table 4 presents the average performance of the algorithm (over ten runs), corresponding to strings generated in the non-dominated Pareto front, for the high-dimensional *Isolet* and the microarray *Colon* cancer data. The algorithm was run for 100

Table 2
Performance of selected feature subsets, of medium cardinality, from Pareto-optimal front.

Dataset	Feature subspace	f_2 (card.)	Validation accuracy (%) by				Silh. Stat. (S)	F-meas.	J	Rand	VI
			k-NN		NB	SVM					
			k =								
Spambase N = 4601 n = 57 c = 2 f_1= 105.8×10^5	{27, 28, 29, 47, 48, 49, 53, 54, 55, 56, 57}	11	1	71.0	61.2	77.4	0.099	0.664	**1.00**	**1.00**	**0.00**
			3	69.9							
			5	70.0							
			7	69.4							
	{3, 4, 9, 10, 11, 22, 23, 24, 35, 36, 37, 38, 52, 53, 54}	15	1	77.9	79.0	**85.5**	**0.131**	**0.742**	0.55	0.58	0.85
			3	78.1							
			5	78.2							
			7	77.9							
	{6, 7, 8, 12, 13, 14, 15, 26, 27, 28, 36, 44, 45}	13	1	**80.4**	65.5	81.2	0.039	0.672	0.82	0.82	0.38
			3	**80.3**							
			5	**80.3**							
			7	**79.7**							
	(Original)	57	1	72.5	**79.3**	83.7	0.098	0.664	–	–	–
			3	72.0							
			5	71.6							
			7	71.1							
Ionosphere N = 351 n = 32 (2–34) c = 2 f_1 = 0.61×10^5	{6, 7, 8, 9, 10, 11, 16, 17, 18, 22, 23, 24, 25, 29, 30, 31}	16	1	90.8	74.9	90.3	0.077	0.724	**0.94**	**0.97**	**0.16**
			3	90.6							
			5	90.2							
			7	90.4							
	{4, 5, 6, 33, 34}	5	1	90.9	88.6	90.6	0.108	0.731	0.54	0.69	0.96
			3	**92.9**							
			5	**92.6**							
			7	**92.3**							
	{4, 5, 6}	3	1	86.2	**89.7**	90.3	**0.156**	**0.836**	0.45	0.57	1.08
			3	90.1							
			5	91.9							
			7	92.3							
	{14, 21, 22, 23, 24, 25, 26}	7	1	**91.7**	70.4	85.8	0.070	0.733	0.75	0.85	0.51
			3	92.0							
			5	92.2							
			7	92.0							
	(Original)	32	1	91.1	81.8	**94.0**	0.078	0.700	–	–	–
			3	91.3							
			5	91.7							
			7	92.0							

The significance of bold values represent the best value in a set of measured values.

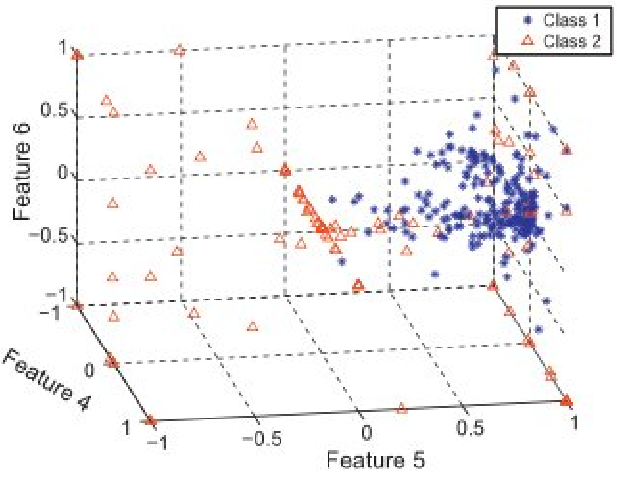

Fig. 6. Projection of *Ionosphere* data in three-dimensional space.

generations with a population size of 40 chromosomes. There were 15,000 generations, with a population size of 200. The 10-fold cross-validation is used to compute the classification accuracy in both the cases.

Table 3
Performance of selected feature subsets, from Pareto-optimal front, allowing variation of clusters.

Dataset	Feature subspace	f_1 (prox. ($\times 10^4$))	f_2 (card.)	Validation accuracy (%) by		Silh. Stat. (S)	F- meas.	J	Rand	VI
				NB	SVM					
Iris N = 150 n = 4 c = 3	{3}	0.549	1	96.7	95.3	0.311	0.933	0.82	0.93	0.37
	{3, 4}	0.556	2	96.0	96.7	0.311	0.950	0.79	0.92	0.42
	{1,2,3,4} (Original)	0.564	4	96.0	96.7	0.311	0.677	–	–	–
Ionosphere N = 351 n = 32 c = 2	{3,5,8, 13,15,17, 19,21,31}	2.739	9	82.6	89.5	0.150	0.740	0.85	0.92	0.29
	{3,5,8, 15,17, 21,31}	2.733	7	86.9	89.7	0.150	0.728	0.85	0.92	0.29
	{3,5,8, 13,15,17, 21,31}	2.737	8	85.7	89.2	0.150	0.746	0.85	0.91	0.29

Table 4
Performance of some selected feature subsets, of large cardinality, from Pareto-optimal front.

Dataset	f_2 (card.)	Validation accuracy (%) by				Silh. Stat. (S)	F- meas.	J	Rand	VI
		k-NN		NB	SVM					
		k =								
Isolet N = 7797 n = 617 c = 26 f_1 = 2.91 $\times 10^7$	275	1 3 5 7	77.6 79.1 80.2 80.7	84.9	94.9	2.4×10^{-3}	0.344	0.88	0.94	0.28
	274	1 3 5 7	77.5 79.1 80.3 80.7	84.8	94.8	2.4×10^{-3}	0.336	**0.89**	**0.94**	**0.26**
	617 (Original)	1 3 5 7	**92.7** **93.7** **94.1** **94.1**	**85.1**	**95.5**	1.5×10^{-3}	**0.365**	–	–	–
Colon N = 62 n = 2000 $n_{preproc}$ = 943 c = 2 f_1 = 1.25 $\times 10^3$	261	1 3 5 7	**83.9** **80.7** 71.0 71.0	54.8	64.5	1.5×10^{-2}	0.704	1.0	1.0	0.0
	264	1 3 5 7	83.9 80.6 71.0 71.0	54.8	64.5	1.5×10^{-2}	0.704	1.0	1.0	0.0
	943 (Preproc.) [2]	1 3 5 7	77.7 79.7 **75.8** **74.8**	53.2	64.5	1.2×10^{-2}	0.704	–	–	–
	2000 (Original)	1 3 5 7	77.1 77.7 75.2 73.9	53.2	**82.3**	2.4×10^{-2}	0.687	–	–	–

The significance of bold values represent the best value in a set of measured values.

With the *Isolet* data we observe that the performance of the classifiers is, in general, better in the original feature space. However, both NB and SVM provide comparable classification accuracy with less than half the number of features. The value of S is found to be better in the reduced space. The values of the J, Rand index and VI index indicate that the clustering obtained in reduced space preserves the structure present in original space.

In case of the *Colon* microarray data we observe that the performance of NB and k-NN (for $k = 1, 3$) is better with reduced features. The same is true for F. Keeping in mind that the reduction in feature set cardinality is almost ten times, as compared to the original set of 2000 features, the overall performance can be said to be reasonably good in the reduced space. For this data, clustering is compared with preprosed feature subset. The J, Rand index and VI index show the proposed FS method preserved the cluster structure in reduced subset. Here clustering is compared with the same in preprocessed feature space.

Table 5
Comparative study on *Iris* data.

Algorithm	Features providing best performance
PR	{3, 4}
DK	{3, 4}
PC	{4, 3}
IM	{3, 4}
*R**	{3, 4}

Table 6
Comparative study with *k*-NN on some data.

Dataset	Algorithm	Accuracy (%)	
		Mean	SD
Iris $n = 4$ $n' = 2, c = 3$	*PR*	94.49	1.34
	BB	92.29	2.57
	SFS	92.29	2.57
	SFFS	92.29	2.57
	SWC	93.48	2.03
	Relief – F	95.68	0.65
Spambase $n = 57$ $n' = 27, c = 2$	*PR*	79.75	0.99
	BB	70.93	0.70
	SFS	70.73	0.77
	SFFS	70.73	0.77
	SWC	76.40	1.05
	Relief – F	89.00	0.28
Ionosphere $n = 32$ $n' = 16, c = 2$	*PR*	78.67	1.81
	BB	75.96	0.35
	SFS	69.94	0.32
	SFFS	74.73	0.37
	SWC	62.03	0.32
	Relief – F	89.90	1.30
Isolet $n = 617$ $n' = 309, c = 26$	*PR*	94.60	0.38
	SFS	74.45	1.20
	SWC	78.25	1.22
	Relief – F	90.40	0.30

3.4. Comparative study

The performance of the proposed algorithm (model PR) for *Iris* data was compared with that of some of the existing techniques, considered as benchmark in this study. These are

1. the statistical method of Devijver and Kittler [10] (model DK),
2. the fuzzy entropy based method of Pal and Chakraborty [28] (model PC),
3. the neural network based method of Ruck and Rogers [36] (model R*), and
4. the model of Ishibuchi [19] (model IM).

Table 5 demonstrates a comparative study of the feature subsets selected by different algorithms for the *Iris* data. As *Iris* data is typically studied by researchers (in the pattern recognition field), an extensive comparison has been provided for this data. The overall study shows that the results tally with each other. The features 3 and 4 are always found to be more important than the features 1 and 2 for classifying *Iris* data.

Next the average performance of algorithm PR was compared (on some of the datasets), using a test set (90% of the data) with certain existing unsupervised techniques, averaged over 10 runs, using a training set size of 10%. The algorithms considered are the

1. branch and bound (model BB) [10]: a search method in which all possible subsets are implicitly inspected without exhaustive search; if the feature selection criterion is monotonic then BB returns the optimal subset,
2. sequential forward search (model SFS) [10]: a suboptimal search procedure where one feature is added at a time to the current feature stage; at each stage the feature to be included in the feature set is selected from among the remaining available features so that the new enlarged feature set yields a maximum value of the criterion function used,

3. sequential floating forward search (model SFFS) [30]: a near-optimal sequential forward search with provision for backtracking, and
4. stepwise clustering (model SWC) [21]: a non-search based scheme which obtains a reduced subset by discarding correlated features.

We also compared the performance of the supervised Relief-F [22] algorithm. Table 6 presents a comparison of the average classification performance, by the k-NN over k = 1,3,5, 7, for sample feature subsets selected by these algorithms for datasets *Iris, Spambase, Ionosphere* and *Isolet*. In each case the initial n features were reduced to n' (as reported in [27], for uniformity of comparison). In general, the proposed algorithm *PR* was better than *Relief-F* for *Isolet* and comparable for *Iris*. As compared to the other algorithms, *PR* was always found to be better.

For *Iris* n' = 2 corresponds to the minimal subset selected by our algorithm in Table 1. However with *Spambase* and *Ionosphere* we observed that an even lower cardinality of 13 (row 4) and 5 (row 6) by *PR* in Table 2 provided a higher classification accuracy as compared to that generated by the larger subsets, n' = 27 and 16 respectively (n' as reported in [27]) in Table 6.

Since *BB* and *SFFS* algorithms required infeasibly high computation time for high-dimensional data, we did not include them for the comparison involving *Isolet*. The performance was best with *PR* for n' = 309 (as reported in [27]). On the other hand, Table 4 indicates the lowest cardinality of 274 with a poorer average classification accuracy (as compared to that using n' = 309). The computational complexity of the proposed algorithm is $O(glnN^2)$ where l is population size and g is the number of the generations. Now complexity of ReliefF algorithm is $O(t_nNn)$ where t_n is the number training sample used for finding nearest neighbor [33]. It has higher time requirement for dataset with large number of samples. The computational complexity of BB, SFS, and SFFS algorithms are infeasibly high for large data set [27].

Interestingly, we also explored the use of c-means [10] clustering during proximity computation. This resulted in the generation of binary values in the proximity matrix, instead of values lying in the range [0, 1]. The presence of a number of "ones" in the matrix perhaps lead to a greater homogeneity between the chromosomes of the population, as evaluated by the first objective function of Eq. (15). Thereby, during multi-objective optimization this objective function plays a less significant role as compared to the cardinality of the feature space (Eq. (16)). Hence c-means almost always results in a minimum cardinality of feature space, typically one, with no emphasis on the cluster structure. This highlights the utility of fuzzy clustering in the algorithm.

4. Conclusion

A new feature selection algorithm, based on structural similarity, has been developed. Fuzzy proximity was used to evaluate the similarity between the original and reduced feature subspaces. The cardinality of the feature subset was simultaneously minimized. The optimal number of features was automatically determined during the multi-objective optimization. This algorithm preserves the performance of the benchmark classifiers as well as cluster structure in reduced space. Comparative study demonstrated the effectiveness of the developed method.

The use of soft computing promises to provide acceptable solutions faster. The topological neighborhood information, pertaining to the inherent cluster structure in the data, is utilized while achieving reduction in feature subspace cardinality. This is expected to have wide ramifications in data mining, data analysis and retrieval, with particular emphasis on visualization.

Although we have restricted our method to numeric attributes, it could be extended to include mixed data by incorporating medoids and considering a symbolic framework for computing the cluster prototypes. This aspect is currently under study.

The basic objective of this paper was to investigate how preservation of structural similarity, as measured by proximity, could help in the selection of appropriate features. Multi-objective genetic algorithm was a tool used during optimization. Any other tool could also have served the purpose. However, in the MOGA framework the size of the chromosome in Eq. (14) is limited by the cardinality n while matrix P is dependent on the number of patterns N. This constrains the algorithm for large data, with a complexity of $O(glnN^2)$. That is one of the reasons why we use preprocessing for the high-dimensional and redundant gene expression data. However, a larger number of patterns can be effectively handled by a divide-and-conquer modularization strategy involving some collaboration amongst independent smaller subsets of patterns. We are currently exploring this scheme for effectively handling larger data.

References

[1] E. Amaldi, V. Kann, On the approximation of minimizing non zero variables or unsatisfied relations in linear systems, Theoretical Computer Science 209 (1998) 237–260.
[2] M. Banerjee, S. Mitra, H. Banka, Evolutionary-rough feature selection in gene expression data, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 37 (2007) 622–632.
[3] A. Ben-Hur, A. Elisseeff, I. Guyon, A stability based method for discovering structure in clustered data, in: Proceedings of Pacific Symposium on Biocomputing, 2002, pp. 6–17.
[4] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[5] Y. Censor, Pareto optimality in multiobjective problems, Applied Mathematics and Optimization 4 (1977) 41–59.

[6] M. Dash, H. Liu, J. Yao, Dimensionality reduction for unsupervised data, in: Proceedings of the Nineteenth IEEE International Conference on Tools with AI, Newport Beach, CA, USA, 1997, pp. 532–539.

[7] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley, London, 2001.

[8] K. Deb, S. Agarwal, A. Pratap, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.

[9] M. Devaney, A. Ram, Efficient feature selection in conceptual clustering, in: Proceedings of the Fourteenth International Conference on Machine learning, Nashville, TN, 1997, pp. 92–97.

[10] P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, Englewood Cliffs, 1982.

[11] F. Douglas, Knowledge acquisition via incremental conceptual clustering, Machine Learning 2 (1987) 139–172.

[12] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, John Wiley, New Jersey, 2001.

[13] R.A. Fisher, The use of multiple measurements in taxonomic problem, Annals of Eugenics 7 (1936) 179–188.

[14] E.B. Fowlkes, C.L. Mallowes, A method to compare two hierarchical clusterings, Journal of the American Statistical Association 78 (1983) 553–569.

[15] A.L.N. Fred, A.K. Jain, Data clustering using evidence accumulations, in: Proceedings of ICPR'02, 2002.

[16] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research 3 (2003) 1157–1182.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, SIGKDD Explorations 11 (1) (2009).

[18] L. Hubert, P. Arabie, Feature selection: evaluation, application, and small sample performance, Journal of Classification (1985) pages 193–218.

[19] H. Ishibuchi, A. Miyazaki, Determination of inspection order for classifying new samples by neural networks, in: Proceedings of IEEE – International Conference on Neural Networks, Orlando, USA, 1994, pp. 2907–2910.

[20] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, 1988.

[21] B. King, Step-wise clustering procedures, Journal of American Statistical Association (1967) 86–101.

[22] K. Kira, L. Rendell, A practical approach to feature selection, in: D. Sleeman, P. Edwards (Eds.), Proceedings of International Conference on Machine Learning, Aberdeen, July 1992, Morgan Kaufmann, 1992, pp. 368–377.

[23] R. Kohavi, G. John, Wrappers for feature selection, Artificial Intelligence 97 (1997) 273–324.

[24] M. Meila, Comparing clusterings – an information based distance, Journal of Multivariate Analysis 98 (2007) 873–895.

[25] S. Mitra, T. Acharya, Data Mining: Multimedia, Soft Computing, and Bioinformatics, John Wiley, New York, 2003.

[26] L.C. Molina, L. Belanche, À. Nebot, Feature selection algorithms: a survey and experimental evaluation, in: Proceedings of ICDM'02, 2002, pp. 306–313.

[27] C.A. Murthy, S. Pradhan, Metric in feature space, in: S. Chaudhury, S. Mitra, C.A. Murthy, P.S. Sastry, S.K. Pal (Eds.), Proceedings of Pattern Recognition and Machine Intelligence, LNCS, vol. 5909, Springer-Verlag, Berlin, 2009, pp. 50–55.

[28] S.K. Pal, B. Chakraborty, Fuzzy set theoretic measure for automatic feature evaluation, IEEE Transactions on Systems, Man, and Cybernetics 16 (1986) 21–41.

[29] W. Pedrycz, V. Loia, S. Senatore, P-FCM: a proximity-based fuzzy clustering, Fuzzy Sets and Systems 148 (2004) 21–41.

[30] P. Pudil, J. Novovicová, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (1994) 1119–1125.

[31] W.M. Rand, Objective criteria for the evaluation of clustering methods, Journal of American Statistics Association 66 (1971) 846–850.

[32] C.J.V. Rijsbergen, Information Retrieval, second ed., Butterworth, London, 1979.

[33] M. Robnik-Sikonja, I. Kononenko, Theoretical and empirical analysis of relieff and rrelieff, Machine Learning (2003) 23–69.

[34] P. Rousseeuw, Silhouette: a graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65.

[35] S. Rovetta, F. Masulli, An experimental validation of some indexes of fuzzy clustering similarity, in: Fuzzy Logic and Applications, Lecture Notes in Computer Science, vol. 5571, Springer, Berlin/Heidelberg, 2009, pp. 132–139.

[36] D.W. Ruck, S.K. Rogers, M. Kabrisky, Feature selection using a multilayer perceptron, Neural Network Computing 20 (1990) 40–48.

[37] L. Talavera, Dependency-based feature selection for clustering symbolic data, Intelligent Data Analysis 4 (2000) 19–28.

[38] H.L. Wei, S.A. Billings, Feature subset selection and ranking for dimentionality reduction, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 162–166.

[39] Z. Zhao, L. Wang, H. Liu, Efficient spectral feature selection with minimum redundancy, in: Proceedings of Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010.